

REMARKS

Applicant requests continued examination in view of the claims as now amended in response to the official action mailed June 10, 2009.

Pending claims 42-47, 49, 52, 53, 55-59 and 61-70 were rejected under 35 U.S.C. §103 over a combination of Britton (US 4,785,408) and Anderson (US 7,240,297). This rejection is respectfully traversed.

As set forth in the official action, the Examiner considers Britton to teach a system wherein a human configuror assembles a dialog to be carried on in a telephone exchange between a processor executing the dialog and one or more remote human subjects. Assembling an operable dialog requires the configuror (the author of the dialog) to define a sequence of dialog components, to choose among options applicable to the components and generally to establish operations and data flow that achieve desired results when the dialog is carried out (executed).

Britton does not teach placing any particular restrictions on what the configuror can define or choose. Britton does not teach the possibility of rendering any assistance to the configuror who composes the dialog. The configuror in Britton is expected to know what he is doing.

Anderson has been cited in combination with Britton. Anderson is an example of an application program operated by a user. Anderson's program has a show-progress feature. The show-progress feature aids the user who is proceeding through a programmed process, by displaying to the user a depiction of the current point in the process, a segment of past activity that brought process up to the current point, and a path that the process is likely to follow in the future, moving forward to its conclusion. It is possible for Anderson to show a likely path of execution in the future, because in Anderson the program is already defined.

Applicant's claimed invention concerns a programmed processor running a procedure that interacts with the configuror, when dialog is being composed. The dialog does not exist until the configuror composes it. The processor is programmed to place constraints on the choices that are made available to the configuror. The

applicable constraints derive from steps taken up to that point, with the aim to force the configuror to define a dialog that is operable, even if the configuror is not proficient and may be substantially ignorant about how dialogs are structured and how dialogs function.

There are two distinct interactive programs involved. One program interacts with the configuror for enabling the configuror to define or author the other program, which other program is the dialog that will interact with remote subjects. Applicant's human configuror may be regarded as the user of the program that assists in composing the dialog. But the configuror is not the user of the dialog. The remote subject is the user of the dialog. One may consider the selections and data provided by the remote subject, such as responses to yes/no prompts, as causing the execution of the dialog to follow one path or another in its programmed sequence. But selections, responses and data from the remote user do not contribute in any way to composing the dialog. The dialog by then has been composed and is set in the stone of program coding.

Anderson's disclosure does not involve authoring, configuring or otherwise composing a program. The input/output of Anderson's program is not with an author who is composing a program. Anderson's disclosure concerns an aspect of an application program that has already been composed and is interacting with an end user. Anderson does not teach or suggest a process for composing a dialog wherein a dialog composition wizard constrains selections offered to a human configuror, who is a dialog developer (a program author) engaged in configuring a dialog that a processor will execute with a remote subject. In Anderson, there is no mention of an author or configuror. That player is absent.

Applicant maintains for a number of reasons stated hereinafter that the situation of an author who is composing a program (such as the configuror of an interactive dialog) is distinctly different from that of an end user who is interacting with an application program that has been composed already (such as a remote subject interacting with an executing telephone dialog or the like). There is no reason to

believe that the disclosure in Anderson of the graphic illustration of past, present and future progress, in Anderson's already-existing sequence of programmed instructions, would be perceived by the person of ordinary skill as having any beneficial parallel in connection with wizard-constrained selection of dialog components when a yet-to-exist dialog is being composed. There is no basis to believe that a combination of Britton and Anderson would render obvious applicant's invention claimed as a whole.

A configuror composing a dialog by selecting and outfitting a sequence of components is free to choose what the dialog will do, according to loops and branches and connections that the configuror sets up. The configuror's choices, made during composition of the dialog, are not branches between alternative paths followed by an executing dialog. The configuror's job is to establish the branches. During execution, the dialog may take alternative paths according to conditions arising during execution of the dialog, such as input from the remote subject, but by then the paths are predetermined and embodied in the previously configured dialog. Any choices that result from input by the remote subject are choices that carry out the pre-configured function of the dialog. There is no authorship or composition involved in executing the dialog.

The configuror's choices are a matter of authorship. Not only are the choices not predetermined, but the function of the dialog is not predetermined either. For this reason, one cannot point to branching paths followed during execution of a dialog (or other program) as being the same as, or even analogous to the choices made by the configuror (or author or programmer) that were originally composed to cause the dialog to behave in a way that achieves a programmed function.

According to applicant's invention as defined in independent claim 42, a processor is programmed to present a dialog composition wizard by which the configuror is offered a selection of dialog components. The configuror selects a sequence of step-by-step instructions by responses entered to the processor via the operator interface. As thereby particularly and distinctly claimed, applicant's assistance is not rendered to a user of an existing program for operating that program.

Applicant's assistance is applied by constraining a configuror who is composing a dialog, i.e., authoring a new program.

The examiner has agreed that Britton does not disclose the aspect of presenting a wizard to guide or assist the configuror in composing the dialog. The configuror in Britton is on his own, and is expected to know all the details necessary to compose an operable sequence of instructions, i.e., to program the dialog.

However, the examiner takes the position that wizards were known at the time of the invention, *per se*. Notably, the examiner cites Anderson as describing a system providing user assistance in utilizing a computer program. Anderson teaches an assistant having the capability to display previous, current and probable future steps during execution of a process, the display providing a form of assistance.

However, Anderson does not provide any basis to conclude that a person of ordinary skill would find any similar sort of assistant to be beneficial with respect to programming or authoring a sequence of instructions so as to compose a program that does not yet exist. Furthermore, if one considers the situation of having an open field of potential choices of dialog components (program steps) and associated options to select, there is no reason to believe that applying Anderson's show-progress idea to selection of Britton's dialog components would constrain the operator in a manner that can be expected to produce operable dialogs.

The examiner's comments in the official action are correct that Anderson's system of user assistance is help for utilizing a computer program. Anderson's assistant does not in fact help or guide or constrain someone who is composing a computer program such as a person configuring an interactive dialog that is to be executed one or more times during interactions with remote human subjects. If the person of ordinary skill sought to apply Anderson's technique for providing user assistance to the Britton system of dialogs, it would be the remote human subject who would be provided with assistance in the form of a display of past, present and future progress in the execution of the already-programmed dialog. Anderson does not teach or suggest a wizard for assistance in configuring an interactive dialog, namely

for composing the sequencing and specification of its program steps. Britton likewise does not teach this aspect.

According to conventional knowledge and the level of ordinary skill, programmers are supposed to know what they are doing when composing programs. Users of already-composed programs are the ones who are provided with assistance. The person of ordinary skill can appreciate the possibility of a show-progress view as in Anderson that shows past progress, the present point and a possible future path, because the program is already in existence and therefore a future path can be estimated. In a situation where a program (in our case a dialog) has yet to be composed, Anderson and Britton, alone or in combination, would not routinely provide the person of ordinary skill with any expectation that one could or should constrain further elements of a dialog (a program) while the dialog is being configured (programmed).

Anderson teaches operating a program where the so-called assistant helps the user with achieving a task that is deterministically defined because the program has already been composed/written. The already-written program embodies behaviors that may do different things for different input values, e.g., from its end user. The user being assisted by Anderson is provided with the ability to test the results obtained or expected using alternative input values. But in the context of Anderson, the program is predetermined. The Anderson program executes from the beginning to the end of a project with defined stages that are not being composed and are not subject to alteration.

The context of applicant's dialog composition is fundamentally different. There is no deterministic programmed sequence or set of alternative potential sequences defined at the time that applicant's configurator is confronted with the task of establishing a sequence of dialog components and choosing among options that are to be applicable to those dialog components. There is likewise no deterministic programmed sequence or set of alternative potential sequences already defined when

Britton's configuror is confronted with this task. In these cases, the dialog has yet to be written.

The difference is fundamental. Anderson's situation is analogous to helping a player of a board game having branching paths already printed on the board. The printing defines all possible paths leading from one or more starting positions to one or more successive positions. The positions and paths are predetermined because the game has already been composed. Player input and game rules may cause the player to follow one potential path or another, but these aspects are predetermined by the printed paths and the rules of the game (they are already programmed). Anderson would assist the player of such a game by highlighting the player's current position on the board, recent history through branches in getting to that position, and a likely future path from the current position to the destination.

But for a dialog configuror, there is no predetermined set of paths. It is the challenge of the configuror of a dialog to define paths and branches by setting up a sequence of connected dialog components. In fact, until the configuror has conceived some operative function for the dialog, there is no destination either. Neither Anderson nor Britton, nor their combination, demonstrates that a person of ordinary skill would consider it possible, let alone probably beneficial, to provide a wizard to guide a configuror in the context of composing or programming a dialog, which characterizes the invention defined in applicant's claims.

Furthermore, applicant's claims do not call merely for some nebulous or undefined form of assistance. Applicant's claims particularly and distinctly define a wizard that offers choices to the configuror and constrains the subsequent choices offered to the configuror as a function of the configuror's responses to previously offered choices.

Effectively, applicant's wizard is analogous to allowing the configuror (like the designer of a board game, not the player of the board game) to blaze arbitrary new trails, along previously undefined pathways and through previously unknown branches, to arbitrary destinations. However applicant constrains the configuring of

pathways. For example, if the configurator defines a branch in the path representing a choice of mutually exclusive alternatives for the remote subject executing the dialog, the configurator can be denied a further option if that would represent an impossible sequence. The wizard constrains the configurator to compose a dialog that will work. It is altogether a different matter to assist a user in traversing previously defined paths.

Applicant's configurator is empowered by the programmed processor to configure a dialog that might have an arbitrary succession of activities, resulting in any number of possible paths, stages, branches, loop-backs, beginning points, ending points, prompts and replies, that the configurator can conceive. However, applicant's programming wizard serves to constrain the configurator during such set up by offering choices that will advance matters, and refraining from offering choices that will fail. By comparison, Anderson's program is already configured. The Anderson assistant in highlighting the past, present and likely future path does not help the operator to compose the program, instead helping the user to execute a program that has already been configured. Anderson is not concerned with the programmer who composed his program.

In previous exchanges between applicant and the examiner, it was mentioned that Anderson has the capability to display paths leading to error conditions. Applicant argued that an ability to test paths including "wrong" paths is the opposite of a constraint as claimed. The examiner's position appears to be that identifying an alternative path as being the "wrong" path is a form of constraint. Applicant does not agree that displaying a wrong path is a constraint if the user is permitted to select that path. But even assuming, *arguendo*, that choosing an alternative path that raises some sort of "error" flag, as in Anderson, reasonably suggests a constraint, there still is nothing in the combination that logically connects to the task of configuring dialogs, i.e., composing programs that have not been written yet and don't have paths to be tested. The cited art supports only the possible benefits of testing among predetermined possible paths in an application program as in Anderson that has been written already and whose paths are set in stone. Anderson's program is only able to

assess the most likely future path because Anderson's "train tracks" have been laid down, and all that remains is whether or not the input from the user of the application program will guide the train along one branching track or another when user input operates the switches. No track is being laid.

Anderson supplies assistance in the execution of a previously existing program. Applicant's dialog system does not exclude the possibility of providing guidance for users, but guidance as in Anderson would mean offering help features for assisting the remote subject in interacting with applicant's already-configured dialog. An interactive dialog as disclosed by applicant advantageously prompts the user, and advises the user what is expected. Such prompting might be used to remind the user about data that has been input up to that current point, possibly to identify the current point, and to advise the user that continuing on the path will lead to some conclusion. Paragraph [0050] of applicant's disclosure mentions how various types of assistants can be used. But applicant addresses a different problem, a problem that Anderson does not even suggest exists, namely assistance in the form of a constraining programming wizard that is active when dialogs are being composed (as opposed to being executed) and at that point might take on any of a wide variety of possible sequences and configurations. Applicant constrains the configurator. The cited references do not.

Interactive telephone dialogs may be simple or complex. For example, a voter-polling dialog might ask remote users for a yes/no signal on a single voter initiative and be done. A more complicated polling dialog might prompt for a choice among plural candidates in an election. A yet more complicated dialog might prompt for a party affiliation, a zip code, a gender selection, then branch on the remote subject's input, and prompt for a choice among a specific subset of candidates. These are arbitrarily selected activities that a configurator may seek to program. It is applicant's invention to enable the configurator to set up paths and branches arbitrarily in a desired sequence to a desired end, while at the same time applying a wizard that constrains the configurator, for example to prevent the configurator from programming an impossibility. This is not the same as Anderson's method of showing the user of an application program (the

remote subject) how he got to the point at which he finds himself by making selections while interacting with an already-programmed sequence.

These distinctions are embodied and defined in applicant's claims, wherein the configurator controls the composing and storing of a dialog that thereafter executed to accomplish input and output operations over a telephone network with a remote subject. Applicant's disclosure involves the creation of interactive dialogs with the help of programming wizard constraint.

The aspects that distinguish over the cited art are particularly and distinctly claimed and are fully supported by the original disclosure. Referring to the paragraph numbering in the published application, applicant discloses the composition of interactive dialogs guiding users (remote subjects) through increasingly complex transactions (para. [0010]), including visually defining interactive dialogs for the benefit of the configurator (para. [0013]) and visually creating dialog structures (para. [0043]). The interactive dialogs may be complex (para [0044]). Applicant discloses a wizard as providing a mechanism for defining how the composed process will flow (para. [0049]). Use of a wizard permits programming constructs, such as sequences, loops and "switches" such as 'if-then-else' constructs (para [0054]) that may vary operations based on input from the remote subject. These constructs are finished by the time that the remote subject (the user) interacts with the dialog. Applicant's wizard applies to the activities of the configurator (author of the dialog), which is not disclosed or suggested by the references cited in support of the rejection under 35 U.S.C. §103.

Anderson does assist the end user of a composed program. For certain situations, assistants of a type taught by Anderson are unavailing or counterproductive. In the situation at issue, where there is no predetermined program yet in existence (namely when composing a dialog), a configurator advantageously should be free to be creative. Yet as taught by applicant, the configurator may not be an expert at composing dialogs. By providing a dialog composing process with a wizard having option constraining capabilities, the configurator is optimally given freedom to be creative without requiring the technical know-how of an experienced programmer. In

many instances, this enables a configuror, such as a business user who is a novice at interactive dialogs, to compose dialogs that the person otherwise would never be able to produce.

The claims as amended are definite. The differences between the invention and the prior art with due regard to the level of ordinary skill, are such that the subject matter claimed as a whole is not shown to have been obvious. The application is in condition for allowance. Applicant requests reconsideration and allowance of pending claims 42-70.

Respectfully submitted,

Date: September 9, 2009

/Stephan Gribok/
Stephan P. Gribok, Reg. No. 29,643
Duane Morris LLP
30 South 17th Street
Philadelphia, PA 19103-4196
tel. 215-979-1283
fax. 215-979-1020
SPGRIBOK@DUANEMORRIS.COM

Docket Nos. U1656-00007